

09/888,607

L Number	Hits	Search Text	DB	Time stamp
1	6235	(test\$6 or analyz\$6 or analis\$6 or diagnos\$6) with memory with controller\$	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 11:27
2	2668	test\$6 adj instruction\$	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 11:32
3	21	((test\$6 or analyz\$6 or analis\$6 or diagnos\$6) with memory with controller\$) with (test\$6 adj instruction\$)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 11:32
4	1	((((test\$6 or analyz\$6 or analis\$6 or diagnos\$6) with memory with controller\$) with (test\$6 adj instruction\$)) with repeat\$6	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 11:33
5	59	((test\$6 or analyz\$6 or analis\$6 or diagnos\$6) with memory with controller\$) with repeat\$6	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 11:34
6	38	((((test\$6 or analyz\$6 or analis\$6 or diagnos\$6) with memory with controller\$) with repeat\$6) and sequenc\$6	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 11:35
7	28	(((((test\$6 or analyz\$6 or analis\$6 or diagnos\$6) with memory with controller\$) with repeat\$6) and sequenc\$6) and register\$	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 12:49
9	153770	repeat\$6 adj3 (cycle\$ or circuit\$ or module\$ or unit or function\$ or instruction\$ or data or register\$)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 12:31
10	127964	repeat\$6 adj2 (cycle\$ or circuit\$ or module\$ or unit or function\$ or instruction\$ or data or register\$)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 12:32
11	20	((test\$6 or analyz\$6 or analis\$6 or diagnos\$6) with memory with controller\$) same (repeat\$6 adj2 (cycle\$ or circuit\$ or module\$ or unit or function\$ or instruction\$ or data or register\$))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 12:32

12	14	4797886.uref.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 12:56
13	2868	repeat\$6 adj2 (instruction\$ or module\$)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 12:59
14	2	((test\$6 or analyz\$6 or analis\$6 or diagnos\$6) with memory with controller\$) same (repeat\$6 adj2 (instruction\$ or module\$))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 12:58
15	64	((test\$6 or analyz\$6 or analis\$6 or diagnos\$6) with memory with controller\$) and (repeat\$6 adj2 (instruction\$ or module\$))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 12:58
16	34137	repeat\$6 adj2 (instruction\$ or module\$ or cycle\$)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 12:59
17	181	((test\$6 or analyz\$6 or analis\$6 or diagnos\$6) with memory with controller\$) and (repeat\$6 adj2 (instruction\$ or module\$ or cycle\$))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 13:00
18	19538	sequence\$ adj3 instruction\$	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 13:01
19	24	((test\$6 or analyz\$6 or analis\$6 or diagnos\$6) with memory with controller\$) and (repeat\$6 adj2 (instruction\$ or module\$ or cycle\$))) and (sequence\$ adj3 instruction\$)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 13:01
20	21	((test\$6 or analyz\$6 or analis\$6 or diagnos\$6) with memory with controller\$) and (repeat\$6 adj2 (instruction\$ or module\$ or cycle\$))) and (sequence\$ adj3 instruction\$) not (((test\$6 or analyz\$6 or analis\$6 or diagnos\$6) with memory with controller\$) with repeat\$6) and sequenc\$6) and register\$) or (((test\$6 or analyz\$6 or analis\$6 or diagnos\$6) with memory with controller\$) same (repeat\$6 adj2 (cycle\$ or circuit\$ or module\$ or unit or function\$ or instruction\$ or data or register\$))))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 13:01

09/888,607

L Number	Hits	Search Text	DB	Time stamp
1	6235	(test\$6 or analyz\$6 or analis\$6 or diagnos\$6) with memory with controller\$	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 11:27
2	2668	test\$6 adj instruction\$	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 11:32
3	21	((test\$6 or analyz\$6 or analis\$6 or diagnos\$6) with memory with controller\$) with (test\$6 adj instruction\$)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 11:32
4	1	((((test\$6 or analyz\$6 or analis\$6 or diagnos\$6) with memory with controller\$) with (test\$6 adj instruction\$)) with repeat\$6	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 11:33
5	59	((test\$6 or analyz\$6 or analis\$6 or diagnos\$6) with memory with controller\$) with repeat\$6	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 11:34
6	38	((((test\$6 or analyz\$6 or analis\$6 or diagnos\$6) with memory with controller\$) with repeat\$6) and sequenc\$6	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 11:35
7	28	(((((test\$6 or analyz\$6 or analis\$6 or diagnos\$6) with memory with controller\$) with repeat\$6) and sequenc\$6) and register\$	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 12:49
9	153770	repeat\$6 adj3 (cycle\$ or circuit\$ or module\$ or unit or function\$ or instruction\$ or data or register\$)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 12:31
10	127964	repeat\$6 adj2 (cycle\$ or circuit\$ or module\$ or unit or function\$ or instruction\$ or data or register\$)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 12:32
11	20	((test\$6 or analyz\$6 or analis\$6 or diagnos\$6) with memory with controller\$) same (repeat\$6 adj2 (cycle\$ or circuit\$ or module\$ or unit or function\$ or instruction\$ or data or register\$))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 12:32

12	14	4797886.uref.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 12:56
13	2868	repeat\$6 adj2 (instruction\$ or module\$)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 12:59
14	2	((test\$6 or analyz\$6 or analis\$6 or diagnos\$6) with memory with controller\$) same (repeat\$6 adj2 (instruction\$ or module\$))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 12:58
15	64	((test\$6 or analyz\$6 or analis\$6 or diagnos\$6) with memory with controller\$) and (repeat\$6 adj2 (instruction\$ or module\$))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 12:58
16	34137	repeat\$6 adj2 (instruction\$ or module\$ or cycle\$)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 12:59
17	181	((test\$6 or analyz\$6 or analis\$6 or diagnos\$6) with memory with controller\$) and (repeat\$6 adj2 (instruction\$ or module\$ or cycle\$))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 13:00
18	19538	sequence\$ adj3 instruction\$	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 13:01
19	24	((test\$6 or analyz\$6 or analis\$6 or diagnos\$6) with memory with controller\$) and (repeat\$6 adj2 (instruction\$ or module\$ or cycle\$))) and (sequence\$ adj3 instruction\$)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 13:01
20	21	((test\$6 or analyz\$6 or analis\$6 or diagnos\$6) with memory with controller\$) and (repeat\$6 adj2 (instruction\$ or module\$ or cycle\$))) and (sequence\$ adj3 instruction\$) not (((test\$6 or analyz\$6 or analis\$6 or diagnos\$6) with memory with controller\$) with repeat\$6) and sequenc\$6) and register\$) or (((test\$6 or analyz\$6 or analis\$6 or diagnos\$6) with memory with controller\$) same (repeat\$6 adj2 (cycle\$ or circuit\$ or module\$ or unit or function\$ or instruction\$ or data or register\$))))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 13:10

21	19	((test\$6 or analyz\$6 or analis\$6 or diagnos\$6) with memory with controller\$) with repeat\$6) and (71\$/\$.ccis.)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 13:11
22	7	((test\$6 or analyz\$6 or analis\$6 or diagnos\$6) with memory with controller\$) with repeat\$6) and (36\$/\$.ccis.)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/03/26 13:11

US-PAT-NO: 5923784

DOCUMENT-IDENTIFIER: US 5923784 A

TITLE: Analyzer and methods for detecting and
processing video data types in a video data stream

----- KWIC -----

Brief Summary Text - BSTX (34):

The controller in the video data stream analyzer includes (i) a statistical processor coupled to receive the parameters from the statistical analyzer and to a plurality of registers, and (ii) a state machine coupled to the statistical analyzer, to the reordering memory, to an output counter, and to the plurality of registers. The state of a bit in each of the plurality of registers represents a characteristic of a field in the reordering memory. Specifically, one register in the plurality of registers is a repeated field register and a bit in the repeated field register indicates, in one embodiment, whether a field in the reordering memory is a repeated field. One register in the plurality of registers is a scene cut register and a bit in the scene cut field register indicates whether a scene cut occurs at a field in the reordering memory. One register in the plurality of registers is a scan pattern register and a bit in the scan pattern register indicates whether a field pair in the reordering memory is progressive or interlaced. One register in the plurality of registers is an odd-even compare register.

US-PAT-NO: 5909404

DOCUMENT-IDENTIFIER: US 5909404 A

TITLE: Refresh sampling built-in self test and repair circuit

----- KWIC -----

Detailed Description Text - DETX (20):

Data retention is measured by a data retention test. State machine controller 212 executes a data retention test by writing to a dynamic memory cell, pausing, and subsequently reading the dynamic memory cell. The data written to the cell and the data returned from the cell are compared to determine data retention or non-retention, equality denoting data retention, and inequality denoting non-retention. In order to determine the cell failure time, state machine controller 212 repeatedly performs the data retention test with changing values for the pause time. For example, if a first data retention test reports data retention, the pause times for subsequent data retention tests may be successively increased until non-retention is detected. The transition between retention and non-retention identifies the cell failure time. In addition, if a first data retention test reports data non-retention, the pause times for subsequent data retention tests may be successively decreased until data retention is detected. Again, the transition between retention and non-retention identifies the cell failure time. In one embodiment, during the data retention tests, state machine controller 212 performs weak write operations and normal reads. Since a weak write deposits less charge into a cell than a normal write, the cell failure time is advantageously decreased. This translates to a decreased overall time to determine the failure times for the sample subset.

US-PAT-NO: 6651201

DOCUMENT-IDENTIFIER: US 6651201 B1

TITLE: Programmable memory built-in self-test combining
microcode and finite state machine self-test

----- KWIC -----

Abstract Text - ABTX (1):

A finite state machine (FSM) is used to generate, in real time, potentially long sequences of signals which control generation of signals for application to a memory structure during a self-test procedure which is provided in hardware on the same chip with the memory structure. The FSM-based instruction generator requires much less area than is required for storage of a corresponding number of microcode instructions and allows the built-in self-test (BIST) controller to have a modular architecture permitting re-use of hardware designs for the BIST arrangement with consequent reduction of elimination of design costs of the BIST arrangement to accommodate new memory designs. The sequential nature of the operation of a finite state machine as it progresses through a desired sequence of states is particularly well-suited to controlling capture of signals where access to high speed data transfer circuits cannot otherwise be accommodated.

Brief Summary Text - BSTX (13):

The memory BIST controller 20 includes a microcode-based controller and instruction decode logic in order to reduce the amount of storage required to provide the desired test signals in accordance with a memory test algorithm which is described in terms of a set of supported instructions stored in an instruction storage module 40 within controller 20. The size of the instruction store module 40 thus depends on the number of required instructions and constitute the largest contribution to area overhead of the BIST unit.

Brief Summary Text - BSTX (14):

The process flow within programmable BIST controller 20 and its components is illustrated in FIG. 3 (which is also not admitted to be prior art as to the

present invention). Once the test algorithm is designed and compiled as indicated at 30, 31, the programmable BIST controller is initialized with a set of instructions representing the selected test algorithm, 32, for example, through an external tester. An initial instruction is dispatched (33) to the instruction decode logic 35 unless it is the last instruction, as determined at 34 to exit the testing process. The instruction is decoded into one or more test signal patterns which are applied to the memory in sequence while the responses are collected and possibly evaluated. Then the next instruction is fetched or dispatched and the process is repeated until all instructions for generation of test patterns have been executed.

Brief Summary Text - BSTX (15):

Testing of complex embedded multi-port memory structures requires an especially complex test algorithm which, in turn, requires a large number of instructions and a large microcode-based controller which, in general, cannot be provided within the 2% area overhead constraint alluded to above. Therefore, programmable BIST architectures are not efficiently applied to complex multi-port memory structures. The only alternative to the area overhead would be to reduce thoroughness of the test procedure which is not feasible due to the high reliability of the memory structure which must be assured.

Brief Summary Text - BSTX (16):

Further, it should be appreciated that the design effort and cost of developing and compiling large instruction sets for programmable BIST architectures capable of testing multi-port complex memory structures and the design of BIST circuits incorporating them on a chip greatly increases chip design costs. This additional cost can be particularly appreciated since it is an underlying goal of programmable BIST circuits to permit accommodation of different circuits to be tested without redesign of the BIST controller and peripheral devices 30 such as are illustrated in FIG. 2 and which may be regarded as respective portions of the instruction decode logic for respective signals applied to the memory under test. This advantage of avoiding a need for custom hardware design is, of course, lost if the hardware design

must be extensively modified to accommodate additional large numbers of instructions. Further, the provision of instructions in programmable ROM is, itself, somewhat self-defeating since the testing of such programmable ROM is also difficult if the storage space is relatively large.

Detailed Description Text - DETX (18):

If the specified instruction generate module has been activated, an instruction is spawned from the activated instruction generate module in accordance with the current state of the finite state machine (FSM), as shown at 240. One or more instructions may be spawned from any specific state of the FSM. Changes of state of the FSM of the instruction generate module can be controlled in any desired manner including but not limited to comparison of the test results with expected results or a given number of repetitions of an instruction or sequence of instructions; each instruction providing one or more signals for exercising the memory under test. Instructions may also be spawned which control the capture of information at accessible points in the memory structure when the result of the test is not otherwise directly accessible.

US-PAT-NO: 6484282

DOCUMENT-IDENTIFIER: US 6484282 B1

TITLE: Test pattern generator, a memory testing device,
and a method of generating a plurality of test patterns

----- KWIC -----

Brief Summary Text - BSTX (6):

The conventional semiconductor memory testing device is shown in FIG. 1. The conventional semiconductor memory testing device comprises a sequence controller 62 and a pattern former 26. The sequence controller 62 controls the generating order of the test patterns for testing a semiconductor memory device. The sequence controller 62 generates an address signal 102 to be output to the pattern generator 26. The pattern generator 26 generates an address pattern signal 106, a data pattern signal 108, and a read write pattern signal 110. The address pattern 106 is input to address input pins of the memory device. The data pattern signal 108 is a data to be written on the memory device. The read and write pattern signal 110 assigns either a write cycle in which the data of the data pattern signal 108 is written on the memory device, or a read cycle in which the data written on the memory device is read out and compared with an expected signal, which is same as the data pattern signal 108.

Brief Summary Text - BSTX (7):

The sequence controller 62 comprises a vector memory for storing vector instructions which indicate the generating order of the test patterns, a read out controller 14 for reading out the vector instructions from the vector memory 12, a vector cache memory including bank memories 16A and 16C, a multiplexer for selecting either of the bank memories 16A and 16C to output the instructions, and an address expander 22 for generating the address signal 102 based on the instructions input from the pattern multiplexer 20. When

the vector instructions read out from the vector memory 12 are being stored into one of the bank memories 16A and 16C, the vector instructions stored in the other of the bank memories 16A and 16C are read out and input to the address expander 22 via the pattern multiplexer 20.

Brief Summary Text - BSTX (12):

FIG. 3 shows an example of the sequence control instruction stored in the address expander used for generating the address signal 102. The instruction "NEXT" of the address #0 indicates that the instruction of the next address, the address #1 in this case, should be output. The instruction "REPEAT" indicates that the instruction of the current address should be repeatedly output "n" times, and following this the instruction of the next address should be output. The instruction "JNI A n" indicates that the instruction of the address marked with a label "A" should be output "n" times, and then the data of the next address should be output. In the example shown in FIG. 3, the address #3 includes the instruction "JNI A 2", and the address #2 is marked with a label "A". The data from the address #2 is output twice at the address #3, and then the data from the address #4 is output. The instruction "STOP" indicates that the test should be terminated. The address expander generates the address signal 102 in accordance with these sequence control instructions to be output to the pattern former 26.

Brief Summary Text - BSTX (13):

FIG. 4 shows compressed instructions stored in the vector memory 12. The sequence control instructions are extremely large in practical usage, so high speed memory with a large capacity is required to store all of the sequence control instructions. Therefore, the sequence control instructions shown in FIG. 3 are compressed for storage in the vector memory 12 in order to save the capacity of the memory. The compressed instructions shown in FIG. 4 are the same as the sequence control instructions shown in FIG. 3. The sequence control instruction "NEXT" shown in FIG. 3 is omitted and the remainder

of the
sequence control instructions are stored in the vector memory 12 with each
address of the instruction written next to the respective instruction.

Brief Summary Text - BSTX (14):

The compressed instruction "REPEAT 4 #1" stored in the vector memory address
#0 of the vector memory 12 indicates that the sequence control instruction of
the address #1 is "REPEAT 4". The compressed instruction "JNI 2 #3 #2" stored
in the vector memory address #1 indicates that the sequence control instruction
of the address #3 is "JNI 2", and the instruction of the address #2 should be
output twice. The compressed instruction "JNI 1 #5 #2" stored in the vector
memory address #2 indicates that the sequence control instruction of the
address #5 is "JNI 1", and the instruction of the address #2 should be output.
The compressed instruction "STOP #6" stored in the vector memory address #3
indicates that the sequence control instruction of the address #6 is "STOP".

Brief Summary Text - BSTX (15):

FIG. 5 shows instructions transferred from the vector memory 12 to the bank
memories 16A and 16C. The sequence control instructions may include a plurality of loops as shown in FIG. 4. Expanding the plurality of loops into successive instructions may delay the generation of the address signal 102.
Therefore, the read out controller reads out the compressed instructions stored
in the vector memory 12 and expands the read out compressed instructions to be
transferred to the bank memories 16A and 16B. As is understood from FIGS. 4
and 5, the instruction of the outside loop "JNI 1 #5 #2" is converted to a
simple instruction "JMP #5 #2" indicating that the address of the instruction
to be output jumps to the address #2 at the address #5. The instruction of the
inside loop "JNI 2 #5 #2" is converted to two separated instructions. When the
instruction "JMP #5 #2" is input, the address expander 22 outputs the instruction of the address #2. Because the instruction of the address #2 is
"NEXT", the instruction of the address #3 "JNI 2 #3 #2" is output as the
address signal 102.

Brief Summary Text - BSTX (17):

Firstly, the address expander 22 accepts the compressed instruction "REPEAT 4 #1" of the cache memory address #0 input from the bank memory 16A. The address expander 22, then repeatedly outputs the data of the address #1 4 times. The next compressed instruction is "JNI 2 #3 #2", therefore the address expander 22 outputs the data of the address #2 and #3 in order. The address expander then repeatedly outputs the data from the address #2 and #3 twice in accordance with the compressed instruction "JNI 2 #3 #2" of the cache memory address #1 input from the bank memory 16A. The next compressed instruction is "JMP #5 #2", which means that the sequence control instructions of the address #4 is "NEXT". The address expander 22 then outputs the instruction of the address #4 and #5 in order. The address expander outputs the instruction of the memory address #2 in accordance with the compressed instruction "JMP #5 #2" of the cache memory address #2 input from the bank memory 16A. As the sequence control instruction of the address #2 is "NEXT", the address expander 22 outputs the instruction of the address #3 in order. The next compressed instruction is "JNI 2 #3 #2", so the address expander 22 outputs the sequence control instructions of the address #2 and #3 twice. The next compressed instruction is "STOP #6", which means that the sequence control instructions of the address #4 to the address #6 are "NEXT" and address expander 22 outputs the instruction of the address #4 to #6 in order. The test is then terminated.

Brief Summary Text - BSTX (26):

The instructions have been previously stored in the control memories 32 of the pattern formers 26A and 26B so that the desired address pattern signal 106, the data pattern signal 108, and the read and write pattern signal 110 are alternately generated by the pattern formers 26A and 26B. In the normal field of the address control memory 32a is stored an address control instruction obtained by combining two successive address control sequence instructions. For example, when the first address control sequence instruction

"XB<0" and the second address control sequence instruction "XB<XB+1" are combined, the value of the XB register becomes 1. Therefore, the instruction "XB<1" is stored in the normal field of the address control memory 32a.

Brief Summary Text - BSTX (27):

The value of the XB register becomes 2 based on the next two address control instructions "XB<XB+1" and "XB<XB+1", therefore the instruction "XB<XB+2" is stored in the normal field of the address control memory 32a. Similarly, the instructions "XB<XB+1", "XB<XB+1", and "XB<XB+1" are stored in the normal field. Stored in the extended field of the address control memory 32a, are the address control instructions obtained by combining two address control instructions which are not executed in sequential order. For example, in FIG. 8, the seventh sequence instruction "XB<XB+1", should be executed after the eighth sequence instruction "XB<XB" is executed. When these two instructions are executed, the value of the XB register increases by 1. Therefore, the instruction "XB<XB+1" is stored in the address #3 of the extended field of the address control memory 32a. The seventh instruction "XB<XB+1" should be executed after the tenth instruction "XB<XB" is executed. When these two instructions are executed, the value of the XB register increases by 1. Therefore, the instruction "XB<XB+1" is stored in the address #4 of the extended field of the address control memory 32a.

Brief Summary Text - BSTX (29):

The combined address control instructions stored in the address control memory 32a of the pattern former 26A and the address control memory 32a of the pattern former 26B have different instructions. This means that the address control instructions stored in the address control memory 32a of the pattern former 26A should be obtained by combining the first and second address control instructions, and the third and fourth address control instructions of the address control sequence instruction. The address control instructions stored in the address control memory 32a of the pattern former 26B is same as the

first address control instruction of the address control sequence instruction, obtained by combining the second and third address control instructions.

Brief Summary Text - BSTX (34):

The semiconductor memory testing device shown in FIG. 7 is capable of outputting the address patterns at a high speed. However, the control memory 32 of the semiconductor memory testing device shown in FIG. 7 is required to have a large capacity because when the instruction is "REPEAT uneven numbers", an additional instruction "NEXT" is required to be written after the instruction "REPEAT uneven numbers". Furthermore, new control instructions obtained by combining two control instructions to be stored in each of the control memories 32, the sequence control instructions and the compressed instructions have to be designed to correspond to the new control instructions. The pattern program is so large that it was difficult to design the compressed instructions in consideration of the new control instructions.

US-PAT-NO: 6002878

DOCUMENT-IDENTIFIER: US 6002878 A

TITLE: Processor power consumption estimator that using
instruction and action formulas which having
average
static and dynamic power coefficients

----- KWIC -----

Detailed Description Text - DETX (56):

Following task 176, control returns to task 72. Decision task 174 causes tasks 170, 176, 72, 80, and process 82 to be repeated for each operating instruction 38 in sequential listing 44.

Claims Text - CLTX (1):

1. A method for estimating power consumed by a processor when performing a sequence of operating instructions, said method comprising the steps of:

Claims Text - CLTX (2):

identifying an operating instruction from said sequence of operating instructions, said operating instruction being associated with one or more actions performed internally by said processor in response to said operating instruction;

Claims Text - CLTX (12):

simulating an execution of said code routine with a processor simulator for said processor, the simulating step providing said sequence of operating instructions;

Claims Text - CLTX (13):

repeating said identifying, obtaining, and computing steps for another operating instruction of said sequence of operating instructions for said processor; and

Claims Text - CLTX (36):

repeating the deriving said instruction coefficients step for other operating instructions for said processor;

Claims Text - CLTX (41):

a controller configured to identify an operating instruction from a sequence of operating instructions, said operating instruction being associated with one or more actions performed internally by said processor in response to said operating instruction, obtain an instruction power formula describing an average static power consumed by said processor for said operating instruction, and obtain an action power formula for each of said one or more actions, each action power formula describing a dynamic power consumed by said processor while executing a corresponding action of said one or more actions;

Claims Text - CLTX (45):

12. A system as claimed in claim 11 wherein the memory further comprises a portion for storing said sequence of operating instructions, said portion having an input for coupling with a processor simulator for simulating said processor and providing said sequence of operating instructions based on a code routine, said code routine being a series of assembly-language instructions for said processor,

Claims Text - CLTX (51):

and wherein the memory further comprises a portion for storing input and output test data for said processor provided by said processor simulator, and wherein the controller is further configured to determine said input and output hamming distances using said input and output test data.

US-PAT-NO: 6675333

DOCUMENT-IDENTIFIER: US 6675333 B1

TITLE: Integrated circuit with serial I/O controller

----- KWIC -----

Detailed Description Text - DETX (23):

The destination of the 8-bit data word is the 8-bit DREG of the target IC.
However, before the 8-bit data word enters to the target IC, it must first be shifted through the bypass bits of ICs 1 through n. To input the 8-bit data word into the DREG of the target IC, the test bus controller outputs control on TMS and TCK signals to cause 108 bits of data to be shifted. After 108 data bit shifts, the 8-bit data word has been shifted through the one hundred bypass register bits of ICs 1 through m and into the 8-bit DREG of the target IC. After the data word is loaded into the DREG of the target IC, the test bus controller outputs control on the TMS and TCK signals to halt the shifting process and load the data word into the memory. This described process of preloading data, shifting data into the target IC, followed by writing the data into the memory, must be repeated for each additional data word written into the memory.

US-PAT-NO: 6640320

DOCUMENT-IDENTIFIER: US 6640320 B1

TITLE: Hardware circuitry to speed testing of the
contents of a
memory

----- KWIC -----

Claims Text - CLTX (1):

1. An electronic system, comprising: a source of test data, which, if the test data source is operating properly, is a pattern of a limited number of data words successively repeated; a memory device for storing the test data; and a memory test circuit having a memory controller coupled to the source of test data and the memory device for controlling access to the memory device, a pattern memory containing the pattern of data words, a controller coupled to the pattern memory and the memory controller for conditioning the memory controller to write test data from the test data source into the memory device and read test data from the memory device and conditioning the pattern memory to read data words from the pattern memory, and a comparator coupled to the memory controller and the pattern memory for comparing the stored test data to successively repeated pattern data words and generating a signal to indicate whether the stored test data is the same as the successively repeated pattern data words; wherein: the controller conditions the pattern memory to repeatedly access successive data words in the pattern of data words and simultaneously conditions the memory controller to read successive locations from the memory device, and responds to the compare signal.

VVV

US-PAT-NO: 6009546
DOCUMENT-IDENTIFIER: US 6009546 A
TITLE: Algorithmic pattern generator

----- KWIC -----

Brief Summary Text - BSTX (12):

Jeffery's system is capable of limited algorithmic vector generation in that it can repeat sequences of instructions. Only the first instance of a repeating vector pattern is stored in the vector memory. Upon encountering that first instance of that pattern during the test, a vector memory controller, in addition to sending the pattern to the pin testing circuits, also saves that instance of the vector pattern in (another) cache memory. Thereafter, when the controller reaches the end of a loop, it starts reading the vectors out of the cache memory instead of the vector memory, and can do so as many times as the pattern "loop" is to be repeated.